筑波大学大学院博士課程

システム情報工学研究科修士論文

手書きによりデザインする 紙製タッチ入力インタフェース

金子 将大

修士 (工学)

(コンピュータサイエンス専攻)

指導教員 田中二郎

2014年3月

概要

現在の計算機環境ではユーザやアプリケーションが多様化しているにも関わらず,計算機の操作はマウスやキーボードなどの汎用的な入力インタフェースによって行われることが一般的である.ユーザやアプリケーションの特徴に合わせた専用の入力インタフェースを用意することができれば,ユーザエクスペリエンスの向上につながると考えられる.しかし,多種多様な要求に既製品の入力インタフェースだけで対応することは困難である.

そこで本研究では、ユーザ自身が自分専用の入力インタフェースを作成できる環境を目指 し、そのアプローチとして紙製の入力インタフェースである紙コントロールパネルを提案し、 その作成システムの実装を行った。紙コントロールパネルは、紙に描いた図形をボタンとし て扱い、図形を指でタッチすることで計算機の操作を行う。紙に図形を描き、アプリケーショ ンの機能のショートカットなどを割り当てることで、タッチ入力インタフェースが作成され る.機能の割り当てはキーイベントやマウスイベントを用いることで簡単に行える。紙コン トロールパネルによって、ユーザは使いやすい入力インタフェースを自作し利用できる。

目 次

序論		1
専用入	.カインタフェース	. 1
本研究の	のアプローチ	. 2
本論文の	の構成	. 2
関連研:	f究	3
手書き)	入力による GUI の設計	. 3
紙を用い	いたインタフェース	. 3
紙とデ	ジタル情報の連携	. 4
操作ウ	ィジェットをレイアウト可能なタンジブルユーザインタフェース	. 5
紙コン	·トロールパネル	6
概要 .		. 6
システ.	ムの使い方...............................	. 6
3.2.1	システムの初期設定	. 8
3.2.2	図形の描画	. 8
3.2.3	図形の登録	. 8
3.2.4	機能の割り当て..............................	. 11
3.2.5	キー操作の割り当て	. 11
3.2.6	マウス操作の割り当て	. 12
3.2.7	テキスト貼り付け操作の割り当て	. 13
3.2.8	登録済み機能の利用・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	. 14
3.2.9	処理コマンドの記述	. 15
3.2.10	記録と復元	. 15
実装		16
初期設定	'定	. 16
各領域	の抽出	. 16
図形の	ラベリング	. 17
機能の	割り当て	. 20
4.4.1	キー操作の割り当て	. 20
4.4.2	マウス操作の割り当て	. 21
	序 専本本 関手紙紙操 紙概シ3.2.1 調用研論 連書をと作 コ要ス.1 3.2.2,3 3.2.5 3.2.7 3.2.8 3.2.1 3.2.1 3.2.7 3.2.8 3.2.1 3.2.1 3.2.7 3.2.8 3.2.1 3.2.1 3.2.1 3.2.2 3.2.1 3.2.1 3.2.2 3.2.1 3.2.1 3.2.2 3.2.1 3.2.2 3.2.1 3.2.2 3.2.1 3.2.1 3.2.2 3.2.1 3.2.2 3.2.1 3.2.2 3.2.1 3.2.2 3.2.1 3.2.2 3.2.1 3.2.1 3.2.2 3.2.1 3.2.2 3.2.1 3.2.2 3.2.1 3.2.2 3.2.1 3.2.2 3.2.1	序調 専用入力インタフェース 本研究のアプローチ 本論文の構成 男連研究 手書き入力による GUI の設計 紙を用いたインタフェース 紙とデジタル情報の連携 繊とデジタル情報の連携 操作ウィジェットをレイアウト可能なタンジブルユーザインタフェース 紙コントロールパネル 概要 システムの使い方 3.2.1 システムの初期設定 3.2.2 図形の描画 3.2.3 図形の置録 3.2.4 機能の割り当て 3.2.5 キー操作の割り当て 3.2.6 マウス操作の割り当て 3.2.7 テキスト貼り付け操作の割り当て 3.2.9 処理コマンドの記述 3.2.10 記録と復元 第 日本 第 日本 キャン 日本 4.1 キー操作の割り当て 4.4.1 キー操作の割り当て 4.4.2 マウス操作の割り当て

	4.4.3 テキスト貼り付け操作の割り当て	21
	4.4.4 登録済み機能の割り当て	22
4.5	指先の認識	22
4.6	機能の実行	23
4.7	紙の情報の保存....................................	23
4.8	ペイント領域の保持と利用..............................	26
4.9	紙の識別	28
第5章	発展	29
5.1	機能の自動割り当て	30
5.2	ウィジェット	31
	5.2.1 トグルスイッチ	31
	5.2.2 スライダ	32
	5.2.3 トラックパッド	33
5.3	紙ならではのインタラクション	33
	5.3.1 紙の変形を伴うインタラクション	33
	5.3.2 複数枚の紙を用いたインタラクション	34
	5.3.3 紙を破棄するインタラクション	34
第6章	利用シナリオ	35
6.1	専用入力インタフェースの作成	35
6.2	データの貼り付け..................................	35
6.3	子供向け玩具	35
第7章	結論	37
	謝辞	38
	参考文献	39

図目次

1.1	紙コントロールパネル	2
3.1	システムの概観	7
3.2	機能割り当て管理用フォーム	7
3.3	ペイントモードによる図形領域の登録前	9
3.4	ペイントモードによる図形領域の登録後	9
3.5	機能割り当て用メニュー	11
3.6	マウス操作割り当ての様子...............................	12
3.7	テキスト貼り付け機能	13
4.1	領域抽出元のカメラ画像	17
4.2	手領域の画像	18
4.3	影領域の画像	18
4.4	図形のラベリング元のカメラ画像	19
4.5	図形の輪郭画像	19
4.6	図形のラベリング画像	20
4.7	正規化前のカメラ画像	25
4.8	正規化後のカメラ画像	25
4.9	元のラベル画像	26
4.10	ペイント画像	27
4.11	ペイント画像を重ねたラベル画像	27
5.1	タッチパネル版画面	29
5.2	ブラウザの機能表..................................	31
5.3	タッチパネル版スライダ	32

表目次

3.1	機能リスト	14
4.1	紙の情報のファイル	23

第1章 序論

本章では本研究の背景として,現在の計算機操作のための入力インタフェースと,その問 題点に対するアプローチについて述べる.

1.1 専用入力インタフェース

計算機操作用の入力インタフェースには、マウスやキーボードをはじめ、さまざまなもの が存在する.入力インタフェースを利用するユーザや操作対象のアプリケーションは無数に 存在するため、それらの特徴に合わせた専用のものが存在すれば、作業効率やユーザエクス ペリエンスの向上につながると考えられる.しかし、これらの多種多様な要求に対して、既 製品の入力インタフェースだけでは対応が難しい.例えば、キーボードのような指で押し込 む操作をする入力インタフェースの場合、手の大きさや利き手、指の太さなどの要因によって ユーザごとに最も扱いやすい形状が異なる.また、一般的なアプリケーションの操作手法の 1つとして、特定のキー操作によって対応する機能を呼び出すショートカットキーがあるが、 キーと操作の対応はアプリケーションごとに異なることが多く、ユーザはそれぞれのショート カットキーの対応関係を覚える必要がある.これは特に初心者のユーザにとって困難である と言える.キーボードとは別に、見た目でどの機能を呼び出すか判断できるボタンを持つ入 カインタフェースが存在すれば、初心者でも扱いやすいと考えられるが、多数のアプリケー ションごとに用意することは難しい.一方で、専用入力インタフェースの自作は、専門の知 識が必要であり、コストもかかるため現実的ではない.

1.2 本研究のアプローチ

本研究では、ユーザ自身が自分専用あるいはアプリケーション専用の入力インタフェース を作成可能な環境を目指し、そのアプローチの一つとして紙とペンを用いて簡単に作成可能 な入力インタフェースである紙コントロールパネルを提案する[1][2]. また、紙コントロール パネルの作成と使用を行うためのプロトタイプシステムを実装した. 図 1.1 に紙コントロール パネルの例を示す.



図 1.1: 紙コントロールパネル

1.3 本論文の構成

第2章では関連研究を挙げ、本研究の方向性や違いを示す.第3章では作成したシステムの持つ機能と使い方について説明する.第4章ではシステムの実装について述べる.第5章では本研究の発展として、得られた意見から考案した実装予定の機能について説明する.第6章では利用シナリオとして本システムの具体的な利用例を挙げる.第7章で本研究についてまとめる.

第2章 関連研究

本章では関連研究について説明し、それらに対する本研究の方向性や優位性を示す.

2.1 手書き入力による GUI の設計

GUI のデザインのスケッチから自動的に GUI を作成する研究が行われている [3] [4] [5]. GUI を作成する際に通常はプログラミングを行う必要があるが,スケッチによるデザインを 可能にすることで,プログラミングの知識がないデザイナがアプリケーションの開発に加わ りやすくなる.

Holzmann らは紙とデジタルペンによりモバイル端末用 GUI のプロトタイプを作成するシ ステムを開発した [6]. デジタルペンによって書かれたスケッチを認識し,インタラクティブ なモバイル端末用 GUI を自動作成する.

Liらはペーパープロトタイピングの様子から自動的にWebページを作成するFrameWireを 開発した[7].Webデザインのスケッチをタッチで操作する様子の動画からリンクの位置や対応するページ遷移を満たすHTMLを自動的に作成する.

本研究は、スケッチすることでインタフェースを作成する点で関連するが、GUI を作成するところからさらに踏み込み、スケッチそのものをユーザが指で触れて操作する入力インタフェースとして扱う.

また、マウスによって操作する GUI のデザインをスケッチによって行う場合、厳密にはマ ウスの動き方や動作範囲などのコンピュータ由来の知識が必要となり、操作の行いやすさを 確認する際は生成した GUI を操作してみる必要がある.一方で、本研究で扱う紙コントロー ルパネルや、モバイル端末の GUI を対象とした [6] の場合はタッチにより操作するため、作 成中のスケッチで行う擬似的な操作と作成後のインタフェースの操作の差異が少ないといえ る.そのため、どのようにデザインすれば使いやすくなるかといった点を考慮しやすく、デ ザインをしながら操作の行いやすさの確認を行うこともできる.

2.2 紙を用いたインタフェース

Keys は紙に描いたウィジェットのスケッチを用いてシンセサイザを操作する SketchSynth を 開発した [8]. ユーザ自身がボタンやスライダ,トグルスイッチなどを描くことで作成し,そ のスケッチ自体を実際に指で操作することができる. 竹川らは紙上に描いた任意形状のオブジェクトに指で触れて音を鳴らす絵楽器のためのプ ロトタイピング支援システムを開発した [9]. 導電性インクを用いて描いた図形に,様々な機 能を持つピンを接続して絵楽器を構築する.

本研究と同様のスケッチを用いた自作可能な入力インタフェースである.本研究では対象 を一般的なアプリケーションに広げている点や,機能の割り当て方法も研究対象としている 点で異なる.

2.3 紙とデジタル情報の連携

Koike らは机上にデジタル情報をプロジェクションすることでデジタル情報と実世界の連携 が可能なシステムを構築した [10]. パン・チルトカメラ,赤外線カメラ,プロジェクタが設置 された机で動作する.赤外線カメラを用いてユーザの手の認識を行う.パン・チルトカメラ はユーザが指差した位置を高解像度で映すように動作する.ユーザが本などに印刷された 2D マーカを指差すことで,パン・チルトカメラがそのマーカを認識すると,プロジェクタが対 応するデジタル情報を机上に投影する.投影されたデジタル情報には手で触れることで操作 可能なものも存在する.

Holman らは紙に対して計算機で表示されるウィンドウを投影し,紙を操作することで情報 を扱うシステム PaperWindows を開発した[11].赤外線カメラとプロジェクタが設置された机 上で動作する.ユーザの指と紙の端に赤外線反射マーカを取り付け,赤外線カメラによって 位置を認識する.計算機のディスプレイに表示されたウィンドウに紙を重ねこするジェスチャ を行うことで,紙にそのウィンドウが投影されるようになる.ウィンドウが投影された紙は タッチディスプレイのようにタップやピンチ操作を行うことができるほか,曲げる,めくる といった紙特有のジェスチャによって操作することもできる.

Akaoka らは実物体に仮想ボタンを設置することができる DisplayObjects を開発した [13]. 赤外線カメラによって実物体や指の認識を行い,プロジェクションで実物体上に投影した仮 想ボタンを操作することができる.仮想ボタンは平面だけでなく曲面にも設置することがで きる.

紙を扱うことで計算機上のデジタル情報を扱うという点で本研究と関連する. どちらも作 業を行う机に赤外線カメラやプロジェクタを設置してシステム環境を構築している.一方,本 システムはエンドユーザが用意しやすいシステム環境である必要があるため, USB カメラの みで動作し,指や紙にマーカなどを設定しない簡潔なシステム環境となっている.また, USB カメラだけであればラップトップ PC とともに持ち運ぶことができるため,紙の利便性の1つ である携帯性の高さを生かすことができる.

Do-Lenhらはマルチユーザ、マルチフィンガを用いたインタラクションにより紙とデジタル 情報を連携して扱うための拡張デスクトップシステム、Decklampを開発した[12]. Decklamp はプロジェクタ、カメラ、ミニPC、タッチ認識のためのレーザ照射装置により構築された持 ち運び可能なプロジェクタ・カメラシステムである.ユーザはマーカによってデジタル情報 が付加された紙に対して、指先やペン先を用いたジェスチャを行うことでデジタル情報にア クセスすることができる.移動可能なシステムを目指した先行研究であり、本研究でも同様 の機能を持つシステムを目指している.本研究では、あらかじめ用意された専用の紙を扱う だけでなく、ユーザがその場で新たな紙の作成や拡張が可能なシステムを目指している.

2.4 操作ウィジェットをレイアウト可能なタンジブルユーザインタフ ェース

物理的に操作可能なウィジェットを用いて計算機の操作を行うタンジブルユーザインタフェースの研究が多く行われている.

Weiss らはテーブルトップ上に設置されたアクリル製のボタンやスライダなどの操作ウィジェットを用いるシステム SLAP Widgets [14] を開発した.テーブルトップの内部から表面を 赤外線カメラで映しており,テーブルトップに接したユーザの指や操作ウィジェットの設置面 に付けられたマーカを認識する.ユーザのタッチやウィジェットによりデータの操作が行わ れ、操作内容がテーブルトップにプロジェクションされる.

Jansen らはタブレットに物理的なスライダを設置することで、大画面ディスプレイのコントローラにするシステムを開発した [15]. タブレットにスライダを設置すると、それに沿った形で専用のメニューが表示される. そのメニューからを大画面ディスプレイを操作するための機能を選択し、実際にスライダを動かすことで大画面ディスプレイを操作することができる.

本研究とは、自由なレイアウトで入力インタフェースを構築することを目的としている点 で関連している.本研究はエンドユーザが使うことを想定しており、専用のデバイスを必要 としない.また、紙を入力インタフェースとすることで携帯性の高さや複製の容易さなどの 利点を持つ.

第3章 紙コントロールパネル

本章では、本研究が提案する紙を用いた入力インタフェース、紙コントロールパネルと、その作成と利用のためのシステムついて説明する.

3.1 概要

紙コントロールパネルは、紙にボタンとする図を描き、機能を割り当て、指でタッチする ことで、割り当てた機能が実行され計算機を操作することができる入力インタフェースであ る.また、計算機を経由してネットワークで接続された家電などの操作も可能となる.

紙コントロールパネルは一般的な入力インタフェースにはない以下のような紙特有の特徴 を持つ.

- 切り貼りや折り畳みによるサイズ変更が可能である
- 薄さと軽さによって高い携帯性を持つ
- 印刷・コピーによる複製が可能である
- 不要になった際の破棄が容易である

紙コントロールパネルは無数に存在するユーザやアプリケーションそれぞれに対応するため,ユーザ自身が必要に応じて新たに作成,カスタマイズすることができるようになっている.

3.2 システムの使い方

本研究では紙コントロールパネルを作成し使用するためのプロトタイプシステムを作成した.本システムは紙とペンの他に,紙を映すように設置された USB カメラを用いる.また,紙へのタッチの認識に手の影を用いるため,ライトスタンドなどの光源が紙の上部にあることが望ましい.カメラは机上の紙を映すように設置する.図3.1 にシステムの概観を示す.

図形への機能の割り当ては管理用のフォームアプリケーションを用いて行う.図 3.2 に機能 割り当て管理用フォームを示す.フォーム中央にはカメラ画像が表示されている.画面上部 のテキストボックスやカラーボックスなどを用いて機能の割り当てを行う.

次に紙コントロールパネル作成の流れとして、図形の描画と機能の割り当てについて述べる.



図 3.1: システムの概観



図 3.2: 機能割り当て管理用フォーム

3.2.1 システムの初期設定

始めて利用する場合,照明環境の違いに対応するため,起動時に抽出領域の閾値を決定す るための初期設定を行う.

フォームに表示されるメッセージにしたがって、紙を画面中央に置き、スペースキーを押 す.紙の設定が完了すると次に手を画面中央に置くように指示されるので、それに従う.手 が認識され、手領域の抽出画像が表示されることを確認後、スペースキーを押す.最後に紙 から手を浮かせ、カメラに手の影が映るようにする.手のときと同様に、手の影の領域が認 識され抽出画像が表示されることを確認後、スペースキーを押す.いずれの場合も、本来認 識されるべきだが認識されなかった座標を左クリック、認識されないべき座標を右クリックす ることで、認識範囲の修正を行うことができる.以上でシステムの初期設定が完了となる. 1度設定した設定はファイルに保存され、読み込むことで復元できる.

3.2.2 図形の描画

まず,紙にペンでボタンとして扱う図形を描く.図形を描く場所や,形,大きさは自由で あり,図形の内部にはメモや書き込みことが可能である.これにより,図形の外見に意味を 持たせて,割り当てられた機能が一目で分かるインタフェースを作成可能である.

図形の登録の際に操作が簡単になるため、四角形や円のような閉じた図形が推奨される.

3.2.3 図形の登録

メニューバーからボタンに用いる色を選択し、カメラプレビュー上で図形を左クリックす ると、選択された図形の領域に色がつき、ボタンとして登録される.この方法によって図形の 領域指定を行う場合は、内部と外部の区別がはっきりしている閉じた図形である必要がある.

図形の割り当てられる色は、メニューバー上のカラーボックスを右クリックすることで表示されるカラーチューザによって変更することができる.

図 3.2 では4つの登録された図形に色がつけられている.

以上の方法により登録した領域は、マウス操作を用いて領域を塗り分けるペイントモード によって修正することができる.カラーボックスを右クリックした際に表示されるコンテキ ストメニューからペイントモードに入ることができる.左ドラッグにより図形の領域を追加 し、右ドラッグにより図形の領域を除去する.これにより、左クリックによる図形内部の領 域の登録では扱えないようなドーナツ型や入れ子状になった図形、閉じていない図形、飛び 地がある図形も登録することができる.

ペイントモードの例として図 3.3,図 3.4 を示す.図 3.3 は左クリックによる領域指定だけ では扱えない閉じた図形以外が描かれた紙であり、それに対してペイントモードを用いて領 域指定を行ったものが図 3.4 である. 左クリックによる領域指定の際と同様に、図 3.4 で色が ついている領域がタッチの有効領域として扱われる.



図 3.3: ペイントモードによる図形領域の登録前



図 3.4: ペイントモードによる図形領域の登録後

図 3.4 に表示されている赤い円はペイントモード時にマウス位置に表示されるマーカであ り、ドラッグの際にこの赤い円の範囲が領域に追加、除去される. 左上の図形は内部の四角形 をタッチの有効領域から除かれたドーナツ型のボタンとする意図で描かれた図形である. し かし、左クリックによる領域指定の際、システムは図形内部は全てメモとして扱うため、内 部の四角形もタッチの有効領域に含む. ペイントモードで右ドラッグにより内部の四角形の 領域を除去することで、図 3.4 のようなドーナツ型の領域を指定することができる. 同様に、 右上の図形では左クリックによる領域指定では指定できない入れ子状の配置を行っている.

左下の図形は閉じていない線から構成されているため、内部の領域を持たず左クリックに よる領域指定が正しく動作しないが、ペイントモードを用いることで自由に領域を指定する ことができる.また、右下の2つの四角形のような飛び地を持ったつながっていない図形を1 つの図形として登録することもできる.この場合、どちらの四角形をタッチしても同じ機能 が呼ばれる.

3.2.4 機能の割り当て

登録済みの図形を右クリックすると機能割り当て用のコンテキストメニューが表示される. メニューにはキー操作とマウス操作のほか,あらかじめ登録された機能が表示される.図3.5 に右クリックメニューの例を示す.メニュー最上部には選択された図形に対応する色が表示 され,以降キー操作・マウス操作割り当て用の項目,その他のあらかじめ登録された機能が 続く.

GUI 上部のテキストボックスには、選択された図形に割り当てた機能のコマンドが表示される.テキストボックスを直接編集することで割り当てる機能を変更することができる.コマンドで指定することができるので、より詳細な機能を割り当てることができる.



図 3.5: 機能割り当て用メニュー

3.2.5 キー操作の割り当て

メニュー選択後に単一のキーと Shift, Control, Alt キーを組み合わせたキー操作を行うことで、その操作を割り当てることができる. 図形へのタッチによりそのキー操作が実行されるようになる. これにより、テキスト入力やショートカットキーを介したアプリケーションの操作が可能になる.

コンテキストメニューから割り当てるほかに、GUI上部のコマンド用テキストボックスを編 集することで割り当てることができる.コマンド上で指定するキー操作を表す文字列のフォー マットには.Net Framework の SendKeys クラスのものを用いた¹.例えば、「A」ならAキーイ ベント、「+{TAB}」ならShiftを押しながらのTabキーイベントが割り当てられる.コマンド で指定する場合は連続的なキー操作を割り当てることもできる.例えば、「ABC」を指定すれ ばAキー、Bキー、Cキーイベントを連続で発生させる.

3.2.6 マウス操作の割り当て

メニュー選択後に操作の対象とするアプリショーンに対してクリックやドラッグを行うこ とで、図形にその操作が割り当てられる.対象となった座標には図形の色と対応したマーカ が表示される.座標は操作対象のアプリケーションの位置を基準として記録されるため、ウィ ンドウの移動を行っても正しく動作する.これにより、GUIを介したアプリケーションの操 作などが可能になる.

例として、図 3.6 にペイントソフトに対するマウス操作を割り当てた様子を示す. 図左側の ペイントソフトのカラーボックスに対する左クリックが 3 つ割り当てられている. 左クリッ クの対象座標に赤,緑,水色の十字のマーカが表示されている. キャンバス部にはドラッグ 操作が割り当てられており,対象座標に青色の矢印が表示されている.



図 3.6: マウス操作割り当ての様子

3.2.7 テキスト貼り付け操作の割り当て

メニュー選択時に表示されるテキストボックスダイアログに文字列を指定すると、その文 字列の貼り付け機能が割り当てられる.図形へのタッチすると、フォーカスしているテキス ト入力領域に対して、設定したテキストの貼り付けを行う.

よく入力する語句を登録しておくことで、辞書と入力補助として使える.

例として、図 3.7 に顔文字貼り付けを割り当てた様子を示す.2つの図形にはそれぞれ別の 顔文字の貼り付けが割り当てられており、図形内部にメモすることで、何が貼り付けられる か一目で分かるように描かれている.図右上に表示されているダイアログによって割り当て を行う.割り当て貼り付けたい文字列を入力し決定ボタンを押すことで、この場合、テキス トボックスに入力された顔文字「(*´ω`*)」の貼り付けが図形に割り当てられる.文字入力 可能な状態で図形にタッチすることで、図左のメモ帳のように顔文字の貼り付けが行われる.



図 3.7: テキスト貼り付け機能

3.2.8 登録済み機能の利用

あらかじめアプリケーションごとの利用可能な機能について定義したリストを用意するこ とで、コンテキストメニューから選択し、その機能を図形に割り当てることができる.リス トは機能の名前・対象のアプリケーション・行われる処理を1セットとし、テキストファイ ルに列挙したものを読み込む.

機能リストを定義することで,連続的なマウス操作や別のアプリケーションの実行など, フォームを介して操作を指定する場合よりも,高度な操作を利用することが容易になる.例 えば,対象のアプリケーションに熟練したユーザが作成した機能リストを不慣れなユーザが 使うことで,途中の操作を意識せずにその機能を利用することができる.また,機能リスト はインタフェースを作成した際に自動で生成されるため,1度割り当てた機能を再度割り当て る場合にも利用できる.

また、図形登録直後に割り当てられるデフォルトの機能も、同様にリストを用意すること で色ごとに指定することができる。例えば、各図形に持たせる機能はあらかじめ決まってお り、デザインを試行錯誤したい場合は、デフォルトの機能としてあらかじめ記述しておくこ とで、図形の登録だけでインタフェースを作り直すことができる。

表3.1に例として、図3.5で用いたペイントソフトの色変更が記述された機能リストを示す. 1行の1つの機能が記述されており、カンマで各項目が区切られている.1行目では「色の変 更:赤」が機能の名前,「#ペイント」が対象のアプリケーションを示す文字列,「LeftClick 455 68」が実行される処理となっている.対象のアプリケーションを示す文字列によってフィルタ リングを行っており、この場合はアプリケーションウィンドウのタイトルに「ペイント」が含 まれている場合のみ、機能が実行されるようになっている.実行される処理の記述はWindows コマンドによって記述されている.この場合は、引数に渡された座標を左クリックするプロ グラム「LeftClick」を呼び出すことで、ペイントソフトの赤色のカラーボックスを左クリッ クするようになっている.

表 3.1: 機能リスト

色の変更:赤, #ペイント, LeftClick 455 68 色の変更:青, #ペイント, LeftClick 566 69 色の変更:緑, #ペイント, LeftClick 521 70

3.2.9 処理コマンドの記述

ボタン登録後または右クリック後に、対象のボタンに登録されている機能を示すコマンド がメニューバー中央のテキストボックスに表示される.これを編集することより詳細な機能 の割り当ても行うことができる.このテキストボックスや機能リストの処理は、コマンドプ ロンプトで用いられる Windows コマンドによって記述する.システムの上でコーディングさ れた機能を呼び出すのではなく、外部の Windows コマンドを呼び出すようになっているため、 マウスやキーの操作では達成できないような複雑な操作を利用したい場合も、実行ファイル などを作成し、コマンドによって呼び出すようにすることができる.

ここで使用するための実行ファイルをいくつか用意しており,例えば,「LeftClick 10 10」と いうコマンドを記述することで,フォーカスが合っているウィンドウの座標系における (10, 10)座標への左クリックを行うことができる.同様にマウス操作用の実行ファイルとして,右 クリックを行う「RightClick」,ドラッグを行う「LeftDrag」「RightDrag」,ホイールの操作を 行う「Wheel」を用意した.

マウス操作以外の基本機能についても、キー操作を行う「SendKeys」、文字列の貼り付け を行う「PasteText」といった形で実行ファイルとして用意されている. コマンドを編集する ことでこれらを組み合わせた操作を呼び出すこともできる.

また、テキストボックスの先頭に#と記号とアプリケーションの名前を入力することで、操 作対象のアプリケーションを指定することができる。例えば「#ペイント」と記述した場合、 アプリケーションのタイトルにペイントという文字列を含むアプリケーションを対象とする。 対象のアプリケーションを指定した場合は、そのアプリケーションにフォーカスが当たって いる場合にのみ機能が実行されるようになる。また、マウス操作の割り当てと同様に、座標 を用いる処理はそのアプリケーションの位置を基準とした相対座標によって動作するように なる。

3.2.10 記録と復元

1度作成した紙コントロールパネルは描いた図形と割り当てた機能が記録される.他の紙を 使用するために入れ替えカメラから外れた後も,再度カメラ前にセットすることですぐに使 用を再開できる.

また、紙コントロールパネルは作業ごとに使い分けられることを想定し、最後に使用して いた際のウィンドウの状態を記録しておき、使用が再開された際にウィンドウの状態も復元 するようにした.これにより紙コントロールパネルを入れ替えるだけで、作業の切り替えを 行うことができる.例えば、ペイントソフト用の紙コントロールパネルを作成しておき、カ メラ前にセットして使用を再開すると、ペイントソフトが起動し前回編集していたファイル が読み込まれる.既に起動していた場合は、ウィンドウを前面に移動しフォーカスする.

第4章 実装

本章では、作成したシステムの実装について説明する.

システムの実装はC#を用いて行った. 各機能はWindows コマンドによって実行される. また, Windows API によってウィンドウの情報の取得やマウス操作の実行を行っている.

OpenCVSharp によって USB カメラの画像を処理することで、図形の認識やタッチの判定 を行っている.本システムで用いられている画像処理として、各領域の抽出、図形のラベリ ング、指先の認識などがある.

システム利用時に行われる処理を順に説明していく.

4.1 初期設定

本システムはカメラ画像を画像処理を行うことで紙,手,影の領域を抽出し,図形やタッ チの認識を行う.その際に,システムが利用される際の照明環境によって認識に用いる閾値 などが変わるため,ユーザの手を借りて初期設定を行う.

まず,カメラ画像の中央に紙が置かれているものとし,矩形の認識を行う.このとき,カ メラが紙の真上に設置されておらず,紙が斜めに映っている場合も許す.ただし,映ってい る紙は必ず長方形であるものとする.

手は HSV 表色系における色相を用いて抽出する.手が画面中央付近に映っているものとし, 手の抽出色相範囲を画像中央付近の領域の色相を含むように設定する.設定した色相範囲に よる抽出画像を表示し,ユーザに抽出範囲が正しいか確認させる.ユーザのカメラ画像に対 する左クリックや右クリックに応じて,抽出色相範囲の修正を行う.左クリックされた座標 を含むように範囲を広げ,右クリックされた座標が含まれないように範囲を狭める.

手の影についても同様に、抽出色相範囲を設定する.

4.2 各領域の抽出

初期設定した各領域の色相範囲を用いて手と影の領域をある程度抽出することができるが, より正確に領域を求めるためにさらに画像処理を行う.

まず,背景差分を求めることで,手と手の影の領域をより正確に求める.手が映っていない状態で背景差分の初期化を行う.手と影の領域は背景差分画像に含まれているので,取得した差分画像に色相による抽出処理を行うことで,より正確に手と影の領域を求めることができる.背景差分の初期化は初期設定終了時と紙が新たに映された場合に自動的に行われる.

最後に膨張縮小処理によるノイズの除去処理を行うことで、手と影の領域を求める. 図 4.1 に元のカメラ画像、図 4.2 に手領域の画像、図 4.3 に影領域の画像を示す.



図 4.1: 領域抽出元のカメラ画像

4.3 図形のラベリング

図形設定用のカラーボックスが選択された状態で、フォームのプレビューが左クリックさ れた場合、紙に描かれた図形の認識を行いボタンとして登録する.カメラ画像から抽出した 紙領域の内部にある図形の輪郭線を求め、クリック座標を含む輪郭の内部をボタンの有効領 域として登録する.登録された図形の領域ごとに塗り分けた二値画像を保持することでラベ リングを行う.1番目の図形の領域の画素値は1,2番目の図形の領域の画素値は2という形 で保持されており、図形が登録されていない領域の画素値は0となっている.

ユーザにもそれぞれの図形の領域を確認できるようにする必要があるが、ラベリング画像 そのものは値の範囲が狭い二値画像であり、見た目で領域を区別することは困難である.そ のため、それぞれの領域をカラーボックスの色で塗り分けなおした画像をプレビュー上に半 透明で表示することで、見た目で図形の領域の区別することができるようになっている.

図 4.4 に元のカメラ画像,図 4.5 に元のカメラ画像,図 4.6 に色つきのラベリング画像を示す.



図 4.2: 手領域の画像



図 4.3: 影領域の画像



図 4.4: 図形のラベリング元のカメラ画像



図 4.5: 図形の輪郭画像



図 4.6: 図形のラベリング画像

4.4 機能の割り当て

図形設定用のカラーボックスが選択された状態で、プレビューが右クリックされた場合、登録された図形への機能の割り当てを行う. ラベル画像におけるクリック座標の画素値を取得し、対応する図形を対象として選ぶ. その後、右クリックメニューから選択されたアイテムやテキストボックスの記述から決定した Windows コマンドと、対象の図形を紐付けて記憶する.

フォームアプリケーション上部のテキストボックスにコマンドが記述された場合は、図形 にタッチした際にそのコマンドが直接呼ばれるようになる.コンテキストメニューから機能 が選択された場合は、それぞれに応じた処理を行って対応するコマンドに変換し、タッチ時 にそのコマンドを実行する.以降、コンテキストメニューから機能が選択された際のそれぞ れの処理について説明する.

4.4.1 キー操作の割り当て

コンテキストメニューからキー操作の割り当てが選択されると、システムはキー操作登録の ための状態に移行する.この状態でフォームアプリケーションの KeyDown イベントが呼ばれ ると、その際に押されていたキーを押す操作をその図形に割り当てる.この際、Shift, Control, Alt のみが押された場合は割り当ての処理を行わず、A キーなどの通常のキーと組み合わせて 押された場合に改めて割り当ての処理を行う.

登録された操作は、キー操作を行う実行ファイル「SendKeys」の呼び出しコマンドとして 保持される.例えば、Shiftを押しながらのTabキーを押した場合は、コマンド「SendKeys +{TAB}」の実行を図形に割り当てることで、キー操作を実現している.

4.4.2 マウス操作の割り当て

コンテキストメニューからマウス操作の割り当てが選択されると、システムはマウス操作 登録のための状態に移行する.この状態では別スレッドを用いてマウスボタンの状態や座標 を監視する.いずれかのマウスボタンが押されるとマウス操作の記録が開始される.マウス 座標が変わらないままマウスボタンが離されるとクリック操作、動いた場合は押した座標か ら離した座標までのドラッグ操作が割り当てられる.これらのマウス座標は、マウスボタン が離された時点でフォーカスがあっているウィンドウを基準とした相対座標に変換して保存 される.また、システムはコマンドとは別に対象アプリケーションとして、このウィンドウ を登録する.

登録された操作は、マウス操作を行う実行ファイル「LeftClick」などの呼び出しコマンド として保持される.例えば、ウィンドウAの相対座標(10,10)を左クリックした場合は、ウィ ンドウAを対象アプリケーションとしたコマンド「LeftClick 10 10」の実行を図形に割り当 てることで、マウス操作を実現している.

マウス操作の実行ファイル群は、与えられた座標に対してマウス操作を行うものだが、こ の座標は実行時にフォーカスが合っているウィンドウを基準とした相対座標として扱われる. 例えば、「LeftClick 10 10」を実行した際に左クリックされるのはウィンドウ座標系における座 標 (10, 10) ではなく、フォーカスが合っているウィンドウ A の相対座標 (10, 10) となる。例え ば、ウィンドウ A がウィンドウ座標系で (100, 0) 座標に存在する場合は、「LeftClick 10 10」で はウィンドウ座標系における (110, 10) 座標が左クリックされることになる。このコマンドの みの割り当てでは、対象アプリケーションでないウインドウ B にフォーカスが合っている状 態で図形がタッチされると、ウィンドウ B の座標を基準とした意図しない「LeftClick」が実 行される。そのため、システム側で対象アプリケーション指定することで、ウィンドウ A 以 外にフォーカスが合っている状態で図形がタッチされた場合は、コマンドが実行されないよ うになっている。

4.4.3 テキスト貼り付け操作の割り当て

コンテキストメニューからテキスト貼り付け操作の割り当てが選択されると、テキストボックスダイアログを表示する.表示されたダイアログに入力されたテキストを貼り付ける操作が図形に割り当てられる.

登録された操作は、テキスト貼り付けを行う実行ファイル「PasteText」の呼び出しコマン ドとして保持される.例えば、ダイアログに「こんにちは」と入力し登録した場合は、コマ ンド「PasteText こんにちは」の実行を図形に割り当てることで、テキスト貼り付け操作を実 現している.

テキスト貼り付け処理はクリップボードを介して行うが、利用前後でクリップボードが変 更されないようにする必要がある.まず、既にクリップボードに保持されているデータを退避 し、貼り付けたいテキスト「こんにちは」をクリップボードに持たせる.次に、WindowsAPI により貼り付けメッセージを発行することで、クリップボードに保持しているテキスト「こ んにちわ」をフォーカスしているテキスト入力領域に貼り付ける.貼り付け処理が完了した らクリップボードを元の状態に戻す.以上の処理によりクリップボードを変更せずにテキス トの貼り付けを行う.

4.4.4 登録済み機能の割り当て

システムは起動時などにユーザによって記述された機能リストを読み込み,コンテキスト メニューに機能割り当てのための項目を追加する.機能リストは表 3.1 に示したように,機能 の名前・対象のアプリケーション・行われる処理を1セットとして機能を列挙したテキスト ファイルである.

機能リストが読み込まれると、コンテキストメニューには機能の名前の項目が表示され、そ の項目が選択されると、対応する機能が図形に割り当てられる.

機能リストには実行されるコマンドと対象とするアプリケーションが記述されている. そ のため特別な前処理は行わず,図形へのタッチ時にはフォーカスが合っているウィンドウが 対象のアプリケーションかどうか確認後,記述されたコマンドがそのまま実行される.

例えば、表 3.1 を読み込んだ場合、コンテキストメニューには「色の変更:赤」、「色の変更: 青」、「色の変更:緑」の3つの項目が追加される.「色の変更:赤」を割り当てた図形をタッ チした場合、フォーカスが合っているウィンドウのタイトルに「ペイント」という文字列が 含まれているか確認後、コマンド「LeftClick 455 68」を実行する.

4.5 指先の認識

抽出した手領域から輪郭線を求め、その輪郭線上に位置する点を一定間隔ごとに取得する. それらの点の中から位置や角度を基に最もそれらしいものを選び指先の座標とする. 同様に 影領域における指先の影の座標も取得する. 図 4.1 では指先の座標に緑の点,指先の影の座標 に青の点が表示されている.

取得した指先の座標と指先の影の座標を用いてタッチ判定を行う.紙から手が離れている 場合,2点の距離は長くなるが,紙に近づくほど短くなる.これを利用し,2点間の距離の変 化から紙と指が接触しているかどうかを判定する.システムは図形が押されている状態かど うかを保持しており,押されていない状態で2点間の距離が閾値を下回れば,図形が押され たと判定しイベントを発生させる.逆に押されている状態で2点間の距離が閾値を上回れば, 話されたと判定しイベントを発生させる.これらの処理に用いる閾値は別々に用意されてお り,2点間の距離が2つの閾値の間に収まる場合は何も処理をしないようにすることで,意図 せずに連続でタッチイベントが発生することを防いでいる.

4.6 機能の実行

図形がタッチされたと判定された場合は、ラベリング画像からタッチ座標の値を取得し、その値に対応した図形に紐付けられたコマンドを実行する準備をする.対象アプリケーションが指定されていない場合は、そのまま実行する.対象アプリケーションが指定されている場合は、Windows API によってアクティブウィンドウを調べ、それが対象アプリケーションならばコマンドを実行する.また、クリック操作などの座標を用いるコマンドの場合は、Windows API によってアクティブウィンドウの位置を調べ、操作対象の座標を再計算する.

4.7 紙の情報の保存

メニューから紙の情報の保存を選択すると、システムはそのとき登録されている紙の情報 として、ラベル画像・図形登録時にクリックした座標・割り当てた機能・対象アプリケーショ ンの状態を保存する.表4.1に紙の情報を記述したファイルの例を示す.改変をしやすいよう に可読性が高い CSV で記述されている.

最初の行は対象のアプリケーションの作業状態を表している.先頭から順に対象のアプリケーションを表すパターンまたは実行ファイルのパス,対象アプリケーションウィンドウの座標とサイズまたは,最大化最小化の状態となっている.これらの情報は紙を再度映した際の識別に用いられる.表4.1の場合は紙を認識した際に,起動中のウィンドウの中からタイトルに「canvas」を含むものを,位置を(100,100)座標,サイズを200×200の状態にして最前面に表示する.

2行目以降は登録された各図形に関する情報になっている. 先頭から順に図形に対応させる色,登録時のクリック座標,割り当てたコマンドとなっている.登録時のクリック座標は 正規化されており,紙の復元の他,マッチングのためのラベリングに用いられる.表4.1の2 行目の場合,図形に対応する色は赤(ff0000),登録時のクリック座標は(445.0,260.0),割り当 てられるコマンドは「LeftClick 460 70」となっている.

表 4.1: 紙の情報のファイル

#canvas, 100, 100, 200, 200
ff0000, 445.0, 260.0, LeftClick 460 70
00ff00, 475.0, 136.5, LeftClick 520 70
0000ff, 225.5, 310.0, LeftDrag 600 220 550 460

ラベル画像とクリック座標は紙領域の四隅の座標を用いて射影変換を行うことで正規化す る.この正規化は、カメラに映っている紙は長方形であるという前提に基づいている.正規 化後のラベル画像は紙を真上から見下ろしたものになり、画像サイズが統一される.正規化 したラベル画像を保存しておくことで、紙の識別の際にカメラと紙の位置関係が情報の保存 時と異なっていても正確に認識できるようになっている.

正規化の例として、図4.7に正規化前のカメラ画像、図4.8に正規化後のカメラ画像を示す. この例では分かりやすいようにカメラ画像を正規化しているが、実際にラベル画像を正規化 したものを保存している.図4.7の正規化前のカメラ画像では、カメラが真上ではなく低い位 置に設置されており、紙が斜めに映っている.図4.8の正規化後のカメラ画像では、は真上か ら撮ったように変換されている.この紙を正規化した場合は、上下や裏表が反転しない限り、 どのように映しても図4.8のようになる.そのため、再認識時にカメラと紙の位置が変わって いても問題ない.

登録時のクリック座標も同様に正規化され保存される.紙の識別時のマッチングのために 行われるラベリングは,正規化した紙の画像に対して,これらの正規化した登録時のクリッ ク座標を用いて行われる



図 4.7: 正規化前のカメラ画像



図 4.8: 正規化後のカメラ画像

4.8 ペイント領域の保持と利用

ペイントモードで追加,除去した領域はラベル画像とは別に正規化して保持される.この ペイント画像は通常のラベリングが行われた画像に対して,再度紙に合わせた変形をして重 ねて描画される.これによりペイントを行った際とカメラと紙の位置が変わっていても問題 なく図形領域の修正が適用される.ペイント画像は紙の情報を保存した際にラベル画像と共 に保存され,復元時に再適用される.

例として,各画像を図4.9,図4.10,図4.11に示す.これらは図3.4で示したようにペイン トモードを利用した際のものである.図4.9は元のラベル画像,図4.10はペイント画像,図 4.11は2つを重ねたものである.ラベル画像とペイント画像は2値画像として保存されるが, ここでは説明のため彩色により領域を区別しやすくしている.

図 4.10 のピンク,緑,水色は追加領域であり、ピンク色の領域は重ねた際にラベル画像の 青色の領域を上書きしている.

白い領域は除去領域であり、重ねた際にラベル画像の赤色の領域を除去している.除去した領域の画素値は0になるため、本来ペイント画像中でも除去領域は画素値0で保存する方が自然である.しかし、0はデフォルトの画素値として処理を加えない領域を表しているため、除去領域は画素値255で扱い、重ねた後で255を0に置換するという処理を行っている.



図 4.9: 元のラベル画像



図 4.10: ペイント画像



図 4.11: ペイント画像を重ねたラベル画像

4.9 紙の識別

同じ紙あるいは別の紙を再度カメラに映すと、システムはその紙が保存された情報と一致 するかを確認する.まず、映された紙の領域を抽出し、四隅の座標から保存したラベル画像 とクリック座標の射影変換を行う.現在映っている紙に変換したクリック座標を用いた場合 のラベル画像を生成し、保存しておいたものと比較する.紙とクリック座標は正規化されて いるため、カメラと紙の位置関係が保存時と異なっていても、正しく比較が行われる.

本システムで扱う図形は特徴が少ないためか, OpenCV で用意されているパターンマッチ などの処理ではうまく動作しなかったため, このような独自の識別方法を実装している.メ ニューから識別方法の変更を行うことで, パターンマッチを用いた識別を行うこともできる.

比較した結果,現在映っている紙と保存してある紙が一致していると判定された場合は,登録された図形や機能,アプリケーションの状態を復元して同じように利用できる状態にする.

第5章 発展

本研究の外部発表を際などに実装したプロトタイプシステムについて、いくつかの意見を 得た.本章では本システムの発展として、それらの意見から考案した機能について述べる.

考案した機能のうちいくつかは、紙を用いたシステムで実装する前に、より実装が容易な タッチパネルを用いたシステムで実装した.タッチパネルを用いることで、紙のシステムと 同じように、図を描いて触れるインタラクションを利用できるため採用した.図 5.1 にタッチ パネルを利用したシステムのスクリーンショットを示す.

ほぼ全ての画面がキャンバス領域となっており、ここに図を描き機能を割り当てることで、 紙のシステムと同様に入力インタフェースを作成することができる.画面左上のボタンにより、描画モードから操作モードを切り替え、その後、描いた図形にタッチすることで割り当 てた機能を実行することができる.画面下にはメッセージを表示する.図 5.1 ではキャンバス 領域に2つの三角形を描き、機能を割り当てている.

システムは Web アプリケーションとなっており,操作対象の計算機上でサーバアプリケー ションを実行し,そこにアクセスすることで利用する.図形がタッチされるとサーバアプリ ケーションを介して,キー操作・マウス操作・外部ファイルの実行などの機能を実行する.



図 5.1: タッチパネル版画面

5.1 機能の自動割り当て

図形に対する機能の割り当てについて、マウス操作キー操作のようにシステム側で割り当 ての補助を行ったものを除いて、コマンドの指定による機能割り当ては分かりにくいという 意見があった.図形の描画はペンで直感的に行うことができる反面、機能の割り当てはフォー ムアプリケーションの操作が必要であり、システムの使い方を理解している必要があったた めだと思われる.

そこでタッチパネル版では、機能が記述された機能リストに対応する図形を設定できるようにした.対応する図形が設定されている場合、その図形を描くだけで対応する機能が自動的に割り当てられる.フォームアプリケーションを操作する必要がある機能の割り当てが省略され、図を描くだけでインタフェースの作成が可能になる.

例えば、ブラウザのショートカットキーを定義した機能リストが存在し、初心者ユーザが リストから作られた図 5.2 のような表を見ながらインタフェースの作成を行った場合、「戻る」 機能を利用したい場合は、対応する図形である左向きの矢印を描けば、それがそのまま「戻 る」ボタンとして動作するようになる.このとき、このユーザが「戻る」のショートカット キーが BackSpace キーであることを知らなくても利用できる.このように機能の割り当てを より簡単に行うことができる.ユーザが対象のアプリケーションに慣れていない場合に特に 効果があると考えられる.図 5.1 では、三角形を描くだけでそれぞれに、前のタブ、次のタブ への切り替え機能が割り当てられる.

タッチパネルでは描いた図について、構成されている線の座標や描いた順番などの情報を 得ることができる.それらの特徴と設定された図形の特徴を比較することで、図形の識別を 行った.図5.1の三角形では、線のなす角度を求めることで三角形であるか、どちらを向いて いるのかを判定した.紙で同様の識別を行う際は、図形の特徴をタッチパネルの場合と同じ ように求めることが難しいため、パターンマッチなどの画像処理を行う必要がある.

インタフェースの作成の際に、行われる処理と対応する図形の紐付けを行うことができる ため、それらに機能の説明を加えることで機能リストを作成することができる.対象のアプ リケーションに熟練したユーザがインタフェースを作成した際に生成される機能リストを、不 慣れなユーザが使うという利用方法を想定している.



図 5.2: ブラウザの機能表

5.2 ウィジェット

作成したシステムでは図形はボタンにあたる動作をするようにした.そのため,割り当て ることができる機能は引数固定の単一のコマンドとなっている.

そこで単一のコマンドの実行では達成できないような機能を扱うための新たなウィジェット について述べる.

5.2.1 トグルスイッチ

本システムをどのように使うか意見を募ったところ、図形をトグルスイッチのように扱い、 押すたびに対象のアプリケーションのモードが切り替わる、設定のオンオフが切り替わると いった機能を割り当てたいという意見が多かった.

現在のシステムでは図形に割り当てることができるコマンドは1つだけであるため、押す たびに動作が変わるようにするには、コマンドで呼び出す実行ファイル側を工夫する必要が ある.

1つの図形に複数のコマンドを設定できるようにし、図形を押すたびに順番に各コマンドが 呼び出されるようにすることで、トグルスイッチのような機能を割り当てやすくなると考え られる.

5.2.2 スライダ

タッチパネル版では、可変の引数を用いたコマンドの実行を行うウィジェットとしてスライ ダを追加した.図 5.3 にタッチパネル版でスライダを設置した画面を示す.スライダは線を引 いた後に、その線の端に閉じた図形を描くことで設置される.閉じた図形がスライダのノブ となり、スライドすることで線の範囲で移動することができる.図で示しているように、ス ライダのノブや線の形状は問わない.

ノブを移動すると、移動量と位置に応じてスライダに割り当てられた機能が実行される.例 えば、スライダにスピーカの音量を調節する機能を割り当てれば、スライダ上で指を動かす ことで音量が変化するようになる.

タッチパネルでは現在値に応じてノブが移動するように実装した.しかし,紙に描かれた ノブを動かすことはできないので,ノブの代替として硬貨などを用いることを考えている.

スライダによって可変引数を扱うことができるようになれば、対象のアプリケーションで スライダのような UI が用いられていた際に、より対応の取れた紙コントロールパネルを作成 することができる.また、線の上に図形を描くというスライダの描画方法は分かりやすく、線 の上で指を動かすという操作方法は直感的であるという意見が得られた.そのため、スライ ダの追加によって、使い方が複雑になりユーザの負担になる可能性は低いと言える.



図 5.3: タッチパネル版スライダ

5.2.3 トラックパッド

本システムは計算機操作のサポートを行うまでにとどまっており、マウスなどの操作と並 行して扱う必要がある.そのためマウスと紙の間を行き来せずに、紙だけで計算機の操作を 行いたいという意見が見られた.

そこでタッチパネル版では、マウスの代替として図形にトラックパッドのような機能を割 り当てられるようにした.図形にタッチした後に指を離さずに動かすことで、指のX方向、Y 方向の移動量を引数として割り当てたコマンドを実行することができる.また、図形に一瞬 だけ触れた場合はタップとみなし、別のコマンドを実行する.これらにマウスカーソルの移 動を行う実行ファイルと、左クリックを行う実行ファイルの呼び出しを割り当てることで、ト ラックパッドのようにマウスカーソルの操作を行えるようにした.マウスカーソルの操作は どのようなアプリケーションでも必要になるため、図形の存在しない領域にタッチしたとき に呼び出される機能とすることも考えている.

紙を用いたシステムでも,指の位置やタッチの認識の精度を高めることができれば,同じ ように実装することができると思われる.

これによりマウスを扱う必要がなくなるため、紙とマウスの間を行き来する必要がなくな り、より使いやすくなると考えられる.

5.3 紙ならではのインタラクション

本システムは紙にウィジェットのスケッチを描き、それをタッチすることで操作を行う.こ れらのウィジェットのデザインや操作方法は GUI で用いられるものを元に実装した.これに 対して、折る、破る、並べるといった紙ならではインタラクションによって操作できると、独 自の用途が生まれるのではないかという意見を得た.

そこで、いくつかの紙ならではのインタラクションについて、その用途と実装方法について検討した.

5.3.1 紙の変形を伴うインタラクション

本システムで扱う紙は長方形であることを前提としている.そのため紙の折り曲げや切断 により、紙が長方形としてカメラに映らなくなると認識を行えなくなる.

これは紙の領域や正規化のために矩形判定を用いているためである。紙の領域については、 白い領域を紙とするなど別の認識方法を用いることで、長方形でない紙も扱うことができる。

正規化については、紙とカメラの位置を固定にする、マーカを用いて紙とカメラの位置関係を認識することで対処することができる。紙に描かれた図形をマーカとして用いることができれば、特に制限されずに現在と同様に扱うことができると思われる.

不定形の紙を扱うことができれば、インタフェースの自由度が増し、より使いやすいデザ インを模索することが可能になる.

5.3.2 複数枚の紙を用いたインタラクション

本システムでは1度にカメラに映る紙は1枚であることを前提としているが、複数枚を同 時に扱えるようにしたいという意見が見られた. 複数の紙を本状に束ね、ページをめくるよ うにして扱うことは可能であるが、複数枚の紙を並べて使うことはできない. 複数枚の紙を 並べて使う利用方法は、タブレットのようなデバイスを用いた際は困難であり、紙を用いる 利点であるといえる.

複数枚の紙の同時認識は、システム側でうまく対応することで現状の認識方法でも可能で あると思われる.

紙コントロールパネルはアプリケーションと1対1で対応させて用いることを想定している. そのためか,紙を机上で動かすことで対応するアプリケーションのウィンドウを動くという機能の提案が多く得られた.

5.3.3 紙を破棄するインタラクション

紙コントロールパネルの利点の一つとして,複製と破棄の容易さをあげた.得られた意見 の中には、使い捨てであることを前提とした利用方法も見られた.例えば、ポテトチップス の油などで手が汚れていて、マウスなどを扱えない場合に用いてはどうかという意見が見ら れた.

その他に、紙を認識することを鍵として、ファイルのアクセスを制限するといった利用方 法が挙げられた.鍵として用いる場合、利用が終了し破棄されたことをシステムが認識する 必要がある.直感的で道具を必要としないため、紙を破くことを破棄のためのインタラクショ ンとして用いようと考えている.

認識方法については、2つに切り裂かれた紙片を認識する、破いた際の音をマイクで認識す るといった方法を検討している.

第6章 利用シナリオ

今回実装したシステムはインタフェースのデザインやボタンに割り当てる機能をユーザが 決めるため、様々な利用方法が考えられる.以下に、システムの利用例を示す.

6.1 専用入力インタフェースの作成

第1章で述べたように本システムにより,扱いやすいボタンの配置で自分のよく使う機能 を実行できるインタフェースを作成できる.ドローイングアプリケーションや3Dモデリング アプリケーションのような,多機能でモードの切り替えなどの操作が頻繁に必要となるアプ リケーションが操作対象の場合にとくに作業効率の向上が見込めると思われる.こういった 場合はショートカットキーを用いることが一般的であるが,本システムはより操作しやすい デザインにすることができ,見た目もどのような機能が割り当てられているのか分かりやす くすることが可能である.

また、紙の画像と機能リストを配布すれば、他のユーザは紙を印刷することで同じように 入力インタフェースとして使うことができる.はじめてそのアプリケーションを使うユーザ に向けて、マニュアルを兼ねた初心者用インタフェースを、アプリケーション作成者や熟練 ユーザが提供するといった利用方法が考えられる.

6.2 データの貼り付け

図形にタッチした際に,文書ファイルや Web ページなどの表示を行うようにすることで, 紙にデータへのハイパーリンクを貼り付けるような使い方をすることができる.

例えば、作業のメモに描いた図や何かを説明する図、既存の印刷物の図などに、参照した Webページや関連する情報が載った文書ファイルなどを開く機能を割り当てることで、後か ら見直す際や他の人が見たときに通常の書き込み以上の情報を付加することができる.

また,作業メモの場合は後で作業を再開した際に,開いていたウィンドウなどの状態を復 元するため,スムーズに作業を再開することができる.

6.3 子供向け玩具

認識にはカメラが必要になるが、作成されるインタフェースは紙であり複製も容易なので、 計算機などの精密機器とは違い、子供に使用させる際に壊すことや怪我をすることの心配を する必要がない.そのため、子供向け玩具の作成に向いていると言える.例えば、既存の絵 本などにタッチする音を鳴らす機能を持たせ、インタラクティブにすることができる.

また、図形の認識により機能の割り当てを自動化できていれば、後は紙に絵を描くだけで インタフェースの作成が可能なるため、子供自身が描いた絵をインタラクティブにする利用 法も考えられる.例えば、動物の絵を描いてタッチするとその動物の鳴き声を鳴らす落書き 帳や、ひらがなを書いてタッチすると発音される文字練習帳などを作ることができる.

第7章 結論

紙でできた入力インタフェースである紙コントロールパネルを提案し、その作成と使用を 行うためのシステムを実装した.ユーザは紙に図形を描きマウス・キーボードマクロをはじ めとした機能を割り当てることで紙コントロールパネルを作成する.図形にタッチすること で割り当てた機能が呼び出され、計算機の操作を行うことができる.

今後はタッチパネル上で実装した機能の自動割り当てやスライダウィジェットの追加など を,紙でも利用できるようにする.また,被験者実験によってユーザビリティなどの評価を 行う.

謝辞

本論文を執筆するにあたり,指導教員である田中二郎先生をはじめ,志築文太郎先生,高 橋伸先生,三末和男先生,嵯峨智先生,Simona Vasilache 先生には,ゼミなどを通して丁寧な ご指導や貴重なご意見をいただきました.心より御礼申し上げます.また,インタラクティ ブプログラミング研究室の皆様には,日頃の研究室での生活を通して多くのアドバイスをい ただきました.深く感謝申し上げます.最後に,ここまで私を支えてくださいました友人や 家族,お世話になった全ての方々に心より感謝いたします.

参考文献

- [1] 金子将大,田中二郎.紙製タッチ入力インタフェース作成システム.第20回インタラクティ ブシステムとソフトウェアに関するワークショップ(WISS2012),日本ソフトウェア科学会, pp.189-190,2012
- [2] 金子将大,田中二郎. 紙コントロールパネル:紙を用いたタッチインタフェースの作成. イ ンタラクション 2014 インタラクティブ発表, 2014(発表予定)
- [3] J. A. Landay, and B. A. Myers. Sketching Interfaces: Toward More Human Interface Design. IEEE Computer, Vol. 34, Issue 3, pp. 56-64, 2001.
- [4] A. Coyette, S. Faulkner, M. Kolp, Q. Limbourg, and J. Vanderdonckt. SketchiXML: towards a multi-agent design tool for sketching user interfaces based on USIXML. TAMODIA '04 Proceedings of the 3rd annual conference on Task models and diagrams, pp. 75-82, 2004.
- [5] V. C. V. B. Segura, S. D. J. Barbosa, and F. P. Simoes. UISKEI: a sketch-based prototyping tool for defining and evaluating user interface behavior AVI '12 Proceedings of the International Working Conference on Advanced Visual Interfaces, pp. 18-25, 2012.
- [6] C. Holzmann, and M. Vogler. Building interactive prototypes of mobile user interfaces with a digital pen. APCHI '12 Proceedings of the 10th asia pacific conference on Computer human interaction, pp. 159-168, 2012
- [7] Y. Li, X. Cao, K. Everitt, M. Dixon, and J. A. Landay. FrameWire: a tool for automatically extracting interaction logic from paper prototyping tests. CHI '10 Proceedings of the 28th international conference on Human factors in computing systems, pp. 503-512, 2010
- [8] B. Keys. SketchSynth: A Drawable OSC Control Surface. Interactive Art and Computation Design 2012, Carnegie Mellon University
- [9] 竹川佳成, 福司謙一郎, Machover Tod, 寺田 努, 塚本昌彦. 絵楽器の設計段階におけるプロ トタイピング支援システムの設計と実装. インタラクティブシステムとソフトウェア XIX: 日本ソフトウェア科学会 WISS2011, pp. 60-65, 2011
- [10] H. Koike, Y. Sato and Y. Kobayashi. Integrating paper and digital information on Enhanced-Desk: a method for realtime finger tracking on an augmented desk system. ACM Transactions on Computer-Human Interaction (TOCHI) Volume 8 Issue 4, pp. 307-322, 2001

- [11] D. Holman, R. Vertegaal, M. Altosaar, N. Troje and D. Johns. Paper windows: interaction techniques for digital paper. CHI '05 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 591-599, 2005
- [12] S. Do-Lenh, F. Kaplan, A. Sharma and P. Dillenbourg. Multi-finger interactions with papers on augmented tabletops. TEI '09 Proceedings of the 3rd International Conference on Tangible and Embedded Interaction, pp. 267-274, 2009
- [13] R. Akaoka, T. Ginn and R. Verteggal. DisplayObjects: prototyping functional physical interfaces on 3d styrofoam, paper or cardboard models. TEI '10 Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction, pp. 49-56, 2010
- [14] M. Weiss, J. Wagner, Y. Jansen and R. Jennings. LAP widgets: bridging the gap between virtual and physical controls on tabletops. CHI '09 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 481-490, 2009
- [15] Y. Jansen, P. Dragicevic and J. D. Fekete. Tangible Remote Controllers for Wall-Size Displays. CHI '12 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2865-2874, 2012